IDENTIFYING SEARCH ENGINE SPAM USING DNS

A Thesis

by

SIDDHARTHA SANKARAN MATHIHARAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2011

Major Subject: Computer Science

IDENTIFYING SEARCH ENGINE SPAM USING DNS

A Thesis

by

SIDDHARTHA SANKARAN MATHIHARAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Dmitri Loguinov
Committee Members,	James Caverlee
	A. L. Narasimha Reddy
Head of Department,	Hank Walker

December 2011

Major Subject: Computer Science

ABSTRACT

Identifying Search Engine Spam Using DNS. (December 2011)

Siddhartha Sankaran Mathiharan, B.Tech., National Institute of Technology Trichy

Chair of Advisory Committee: Dr. Dmitri Loguinov

Web crawlers encounter both finite and infinite elements during crawl. Pages and hosts can be infinitely generated using automated scripts and DNS wildcard entries. It is a challenge to rank such resources as an entire web of pages and hosts could be created to manipulate the rank of a target resource. It is crucial to be able to differentiate genuine content from spam in real-time to allocate crawl budgets. In this study, ranking algorithms to rank hosts are designed which use the finite Pay Level Domains(PLD) and IPv4 addresses. Heterogenous graphs derived from the webgraph of IRLbot are used to achieve this. PLD Supporters (PSUPP) which is the number of level-2 PLD supporters for each host on the host-host-PLD graph is the first algorithm that is studied. This is further improved by True PLD Supporters(TSUPP) which uses true egalitarian level-2 PLD supporters on the host-IP-PLD graph and DNS blacklists. It was found that support from content farms and stolen links could be eliminated by finding TSUPP. When TSUPP was applied on the host graph of IRLbot, there was less than 1% spam in the top 100,000 hosts.

To my parents and sister

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Dr. Dmitri Loguinov for his guidance throughout my research. I thank him for agreeing to be my advisor and for making me a part of the Internet Research Lab. I am thankful to him for the critical feedback and constant motivation during our interactions, which helped me improve my thesis. I would like to thank Dr. A. L. Narasimha Reddy and Dr. Ricardo Bettati for agreeing to be part of my committee and giving their valuable suggestions. I also thank Dr. James Caverlee for agreeing to be part of my committee on short notice and for his suggestions.

I would like to thank the members of the Internet Research Lab and in particular, I thank Sadhan Sood, Tanzir Ahmed and Xiaoxi Zhang for their help during my research. I also thank my colleagues at CSG and my friends in College Station for keeping me cheerful.

I am indebted to my parents for their constant support and motivation throughout my graduate studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Search-engine spammers make use of Search Engine Optimization (SEO) techniques that aim to manipulate ranking algorithms to give spam websites better placement. Users are misled to the spam sites which generate revenues from click-through advertisements. This makes it essential for search-engines to provide reliable results and eliminate spam.

The motivation for our work comes from experiences with IRLbot [1]. When we sampled pages from IRLbot, we found significant amount spam pages from spam farms. Our work aims to identify such spam farms and avoid them while crawling. Commercial search-engines like Google own large server clusters and huge bandwidth to massively crawl the web. Crawlers from research labs and academia lack the bandwidth and computing resources to crawl massively. Hence, it is crucial for such crawlers to efficiently allocate the available resources to high quality websites. We can avoid allocating huge crawl resources to spam sites if we could rank them lower. There is also not much research that is publicly available on the design of web crawlers.

A. Problem formulation

It is challenging to rank infinite resources like hosts and pages as spammers could create a web of hosts and pages to manipulate search-engine ranking algorithms. It is hard to differentiate a spam page and quality website in real-time. There is also limited crawl resources in academia and research projects. Hence, it is necessary to rank hosts in real-time during crawl and allocate budgets.

The journal model is *IEEE/ACM Transactions on Networking*.

B. Contributions

We solve the problem of ranking infinite resources by using finite resources. We use finite resources like Pay Level Domains(PLD) and IPv4 addresses in ranking hosts. We use crawl data from IRLbot for all our experiments.

We first tried ranking hosts using existing algorithms like level-1 supporters(IN), PageRank [2] and level-2 supporters(SUPP) [3] on the host graph. PageRank performed poor and had lot of spam in the top 100 hosts. There were around 10% spam sites in the top 100,000 hosts ranked by PageRank. We found this was because the spam hosts on the top were supported by a huge number of spam hosts with a small outdegree. IN similarly performed poor and had around 17% spam sites in the top 100,000 hosts. While SUPP also had around 9% spam sites in top 100,000, there was only 1 spam site in the top 1000 sites. All of the above algorithms failed because an infinite resource was ranked using an infinite resources. Spammers can manipulate all these ranking algorithms when applied on the hostgraph.

We next ranked IPs using the IP graph which is derived by reducing the host graph of IRLbot. We thought we could find higher quality hosts on higher ranked IPs. We ranked IPs using IN and SUPP on the IP graph and found that there were few spam IPs ranked higher. On further analysis we found that the IPs were parking services owned by hosting providers like GoDaddy. Hosting IPs had high IN and SUPP on the IP graph due to the IP diversity of the sites hosted on them. Hence, we found that the IP graph was not very useful to identify quality hosts.

We also tried to find spam IPs by calculating the host densities of IP addresses. It has been found that a high host density is indicative of spam [4]. We found that 85 of the top 100 IPs based on host densities were completely spam when sampled. The non-spam IPs consisted of blogs and hosting providers.

We propose a novel method of using heterogeneous graphs extracted from the webgraph for ranking hosts. We propose that PLDs can be used to rank various resources by generating the appropriate heterogeneous graph. We first propose PLD supporters (PSUPP) which counts the number of level-2 PLD supporters on the host-host-PLD graph. We found that this was efficient in ranking infinite resources like hosts and had only around 3% spam in the top 100,000 hosts.

We improve PSUPP with a few modifications. We try to find the true PLD supporters to hosts. We define true support as the support that comes from hosts ranked lower than a host. We also remove support from outliers in the host graph when a single host contributes a large share of the total support. We then merge inlinks in the host graph from the same IP to a single randomly chosen link. This removes support from spam farms which have a lot of links emerging from a few IPs. We finally remove blacklisted hosts. We find the blacklisted hosts through two different methods. In the first method, we build a DNS graph using hosts, IPs and authoritative DNS nameservers. We perform a controlled BFS on this graph using the average PSUPP values of an IP address. In the second method, we use the average rank of hosts using PSUPP on a IP address. We remove these blacklisted hosts from the host graph and calculate PSUPP on the updated graph. We call this method True PLD supporters(TSUPP). We found that TSUPP had around 0.3% spam sites on the top 100,000 hosts. We further propose that we can iterate using TSUPP. We can use the TSUPP values to update blacklists and find more spam hosts. After removing the hosts from the updated blacklists, we can calculate the TSUPP values for the next iteration and keep repeating the process. We compare different algorithms using Google Toolbar Ranks (GTR) and manual analysis of sampled sites.

CHAPTER II

RELATED WORK

A variety of spamming techniques can be used against search-engines [5]. Spam farms exploit link based ranking algorithms like HITS [6] and PageRank [2]. Spam farms are used to endorse a target page. They achieve high page scores by interlinking between their pages [7]. A count of level-2 supporters(SUPP) on the ingraph has been found to be effective in eliminating the effect of link farms [3].

There have been numerous solutions to find link spam. TrustRank [8] proposes trust to be propagated through neighbors by starting from a seed set. SpamRank [9] proposes penalty to a page if the PageRank score distributions of in-degree neighbors is suspicious. Topical TrustRank [10] gives trust scores for each topic separately to ensure different topics on the web are covered. CredibleRank [11] first automatically assigns link credibility for all web pages based on distance from known spam pages. This is then used while ranking pages. Truncated PageRank [12] reduces the effect of PageRank of neighbors on a given page. Trust and distrust can be propagated across the graph through links [13]. There have been studies [14][15] which start with a seed set of spam pages. Then they propagate the penalty value to other pages through their links. This penalty value is then considered together with page scores while ranking pages [14]. Spam is also identified by performing random walks from known spam seed sets [15]. Spam Mass [16] for a page is calculated from PageRank scores and by using a set of known good pages.

Spam farms have been found through large strongly connected components in the webgraph, [17], [18]. Links between pages are re-weighted based on densely connected bipartite components found in the webgraph [19]. Dense subgraphs have been found to identify communities in the web [20].

Similarity between URLs, hostnames, content and other properties of a page, [21], [22], have been used to identify spam. Machine learning classification and discrete analysis on directed graphs has been used to find link spam [23], [24]. Classification based on both linking and content has been done to identify spam [25].

All of the above methods try to find link spam using the webgraph. Pages and links in spam farms are dynamically generated. The idea of giving page scores to identify spam is ineffective as new pages and hosts can be created. Moreover, identifying spam from the webgraph is complicated due to its size. We can instead use finite Pay Level Domains(PLD) and IPv4 addresses to rank hosts. PLD rankings have been used to allocate crawl budgets [1] [3]. There have been earlier work [4] [26] [12] [27] that suggest using DNS information. High number of hostnames resolving to an IP address is indicative of spam [4]. They are known to use wildcard DNS entries. This enables them to generate infinite number of hostnames with keywords included in them. Hosting IP addresses have been used as one of the features while classifying spam [26]. In Truncated PageRank [12], a small set of hosts is manually inspected for spam. Connected components in DNS queries have been used to find botnet client-server communication inside a network [27].

CHAPTER III

DATASET

The dataset used in this paper is from IRLbot [1]. The crawl data contains 7,437,281,300 pages with valid HTTP replies. The hostnames from these pages were then extracted. There were 641,982,056 hostnames. The distribution of hostnames over different TLDs is given in Table I. We sampled 615 pages from the crawl randomly and classified them manually to estimate the amount of spam in the crawl. The results of the classification are given in Table II. Throughout this paper, we define spam as pages which contain no meaningful content and excessive use of keywords, links and possibly machine generated content. There were approximately 15 % spam pages in the data set. We believe that this makes our dataset better suited to study spam compared to other crawls like WebBase [28] which have little spam.

A. DNS resolutions

The number of IPv4 addresses is finite. We resolved the list of hostnames from the crawl of IRLbot. We record all information from DNS while performing resolution and use it for generating the DNS graph. We implemented a iterative DNS resolver to collect all DNS information for the list of hostnames we have. We record the path taken, authoritative servers seen for each hostname and most of the information returned from a query. We had to implement our own resolver for two reasons (i) to collect all DNS information related to a hostname (ii) to resolve fast. We did not use BIND for iterative resolutions due to it's limitations on the speed at which it resolves. BIND 9 does recursive resolution synchronously which is blocking [29]. We send and receive queries asynchronously. There are thousands of unresolved outstanding queries at any point of time. We then resolved the list of hostnames from IRLbot. We

Table I. IRLbot TLD distribution.

TLD	% hostnames
.com	27.09
.info	26.48
.net	11.37
.org	8.83
others	26.23

Table II. Randomly sampled pages from crawl.

Type	% of non HTTP 404 pages
Not spam	82.60
Spam	14.80
Adult	2.60

Table III. Results of DNS resolutions choosing all authoritative servers (5 × XEON 2.4GHz).

Month run	Aug. 2010
Duration (hours)	38
Queries per sec (qps)	4,585
Queries sent	641,982,056
Queries resolved	641,982,056
Queries with IPv4 responses	175,482,673
Queries with CNAME records	9,220,725
Unique IPv4 addresses	4,266,486
Unique authoritative servers	788,832
Number of hosts with loops	2,685,973
Number of PLDs in hostnames with IPv4 responses	30,766,107

Table IV. Distribution of Google Toolbar Ranks of hosts with an IPv4 address.

GTR value	Number of hosts
10	65
9	1,578
8	28,515
7	76,003
6	378,719
5	1,201,089
4	3,155,279
3	5,887,305
2	6,572,872
1	5,512,704
0	10,304,095
No GTR	142,258,765

ran the resolver on 5 servers with Intel Xeon processors. Some hostnames have more than one authoritative server. We ask all the authoritative servers. This provides us with exhaustive DNS information for a given hostname and we get all the name servers and possible IPv4 addresses. This information is useful in identifying alliances in a spam farm as we show later. We receive a large number of non-existent domain (NXDOMAIN) replies as the hostnames were obtained from the crawl in 2007. We also mark some hostnames as looping when we visit the same nameserver more than three times in the process of resolving it. The resolution statistics are summarized in Table III. We also found the Google Toolbar Rank(GTR)s for the hostnames that were resolved to an IPv4 address. The distribution of the GTRs is tabulated in Table IV.

CHAPTER IV

DNS COHOSTING

A. Sorting by density

It was seen in the past that a high number of hostnames on a single IP address is indicative of spam [4]. We calculated host densities of the IP addresses in our dataset. We analyzed the content on these servers by picking 100 hostnames uniform randomly from them. We manually checked the content at the document root of the hostname and classified it. We found that all of the servers with host density greater than 2,000,000 were hosting spam when randomly sampled.

After resolving the hostnames from our crawl data, we have web servers $\{i_1, i_2, ..., i_k\}$ and PLDs $\{p_1, p_2, ..., p_m\}$. The set of hosts on IP i is given by $N_H(i)$. We calculate the host density $|N_H(i)|$ for web server i. We similarly calculate PLD density $|N_P(i)|$ for web server i. They are then sorted based on the densities. The % spam on each server is calculated by manually checking 100 randomly picked hostnames from the top IPs in the sorted list.

We classified the type of content on the top 100 IPs based on host density and describe them below.

Parking. Parking services were the most commonly found type of server. They hosted a large number of PLDs.

Keyword spam. These servers hosted only a few PLDs, but had a high host density by the use of DNS wildcards. Most of the hostnames appear machine generated with keywords. For example, *language-study-programs.gov-diet.motorheads.ru*. This also included adult sites that had lot of keywords included into the hostname.

Content spam. These servers contained pages that looked genuine, but all the hostnames on the server looked identical. Some of them generated synthetic content by stealing feeds from other sites.

Affiliate marketing. Affiliate marketing sites includes sites like Clickbank, where each user markets their product using a unique link. The user is given their own subdomain to market different products.

The top 15 IP addresses based on host density and their classification is listed in Table V. The different types of servers found in the top 100 IPs based on host density is listed in Table VI. We found 83 web servers to be hosting 100 % spam based on their high host density. We also observed that, out of the 175,482,673 hostnames that had an IPv4 address, 97,841,969 resided on the top 150 IP addresses by host density and most of them were spam when we sampled. The number of URLs from these spam servers that were crawled by IRLbot are shown in Table VII. These servers contained about 5 % of the original crawled pages. We can avoid a significant amount of spam using host density information.

Although we discovered a lot of spam using host density, there were also false positives like *blogspot.com* and *livejournal.com*, which had a high host density as each user is given a subdomain. There are also hosting services, which usually host a lot of sites together on a single server. We can eliminate these false positives using some form of metric from the webgraph. However, this method will not aid us in eliminating all spam pages as there would still be spam pages located on low host density IPs. We next rank IPs using the IP graph which is derived from the host graph of IRLbot.

Table V. Top 15 IPs based on host density.

i	WHOIS	$\|N_H(i)\|$	$\|N_P(i)\|$	Type	% spam
87.118.118.119	Hostmaster Day	10,160,638	832	Content spam	100
209.85.51.196	ThePlanet.com	9,752,777	7,698	Parking	100
68.178.232.99	GoDaddy	8,282,956	161,390	Parking	100
64.20.60.99	Interserver	5,728,080	284	Parking	100
64.20.60.106	Interserver	5,727,955	43,479	Parking	100
216.8.179.24	Managed Network Systems	4,244,538	233,729	Parking	100
208.73.210.28	Information.com	3,419,804	436,879	Parking	100
67.29.139.153	Level 3 Communications	2,831,333	11	Content spam	100
74.63.153.62	Clickbank.com	2,748,719	1	Affliate Marketing	100
74.63.153.63	Clickbank.com	2,748,719	1	Affliate Marketing	100
82.98.86.166	Sedoparking.com	2,650,987	18,173	Parking	100
66.246.235.42	Unknown	2,526,509	45,174	Parking	100
82.98.86.164	Sedoparking.com	1,961,486	35,074	Parking	100
66.246.235.250	Unknown	1,917,217	10,859	Parking	100
64.95.64.197	NameMedia	1,856,219	388,540	Parking	100

Table VI. Types of servers in top 100 IPs based on host density.

Type	% servers	% spam
Domain parking	59	100
Keyword spam	16	100
Blogs	11	0
Content spam	8	100
Hosting	4	0
Affiliate marketing	2	100

Table VII. Spam URLs in top 100 IPs based on host density.

Total URLs crawled	6,380,051,942
Spam URLs in the top 100 IPs	318,656,719 (5%)

CHAPTER V

RANKING ALGORITHMS

PLD level crawl budgets have been allocated using PLD rankings from the PLD graph[3]. We can similarly rank hosts to allocate site level budgets. We compare different existing ranking algorithms and our proposed algorithms on the host graph of IRLbot, which has 641,982,061 unique sites out of these 117,576,295 were downloaded during the crawl. The PLD graph of IRLbot has 89,652,630 nodes, 33,755,361 were downloaded. We also analyze different algorithms in ranking IPs using the IP graph.

A. Existing algorithms

PageRank [2] is based on random walks on the web graph (V,E). It is calculated as the stationary probability of visiting a page j. It follows how a random surfer will propagate through the outlinks of the current page or teleports to a random page. IN counts the number of inlinks at each node of the graph and excludes self loops.

SUPP represents the number of unique nodes i with a path with shortest distance $d(i, j) = 2$ to j. This is the number of unique neighbors at level-2 of a BFS on the ingraph. SUPP was used to rank PLDs in the PLD graph and has been found to avoid spam in the PLD rankings [3]. It was found that $d(i, j) = 2$ performs optimally in ranking PLDs.

We assign sequential IDs to nodes similar to methods for efficient computation of PageRank [30]. The graph is vertically partitioned with each of the partition having all source nodes, but have only edges with nodes corresponding to that partition. We create k partitions of the ingraph as explained in Algorithm 1. We keep few of the partitions in memory based on available RAM. The number of partition required is $\frac{G}{R}$, where G is the graph size and R is the RAM capacity. We then read through the

ingraph from disk to find level-2 supporters in the partitions in memory. We finally merge the counts across different partitions to find the total SUPP value.

$$SUPP(j) = \sum_{p=1}^{k} \sum_{i=1}^{n_p} 1_{d(i,j)=2}, \tag{5.1}$$

where n_p is the number of nodes in partition p.

Input:indegree graph (V, E), V numbered from 0 to $|V|$, number of partitions k ;
Output:k partitions of graph (V, E);
for $i = 1$ to k **do**
  for each vertex $v \in V$ **do**
    $V_i \leftarrow v$
    for each edge $(v, u) \in E$ **do**
      if $partition(u)$=i **then**
        $E_i \leftarrow (v, u)$
      end if
    end for
  end for
end for

Algorithm 1: Partition graph

B. Host graph

We first ranked hosts using the host graph of IRLbot. We ranked them using IN, PageRank and SUPP. We found that IN and PageRank had a significant amount of spam in the top 1000 hosts, while SUPP had only one spam host in the top 1000. However, SUPP has a lot of spam in the top 100,000. The above algorithms failed because an infinite resource is being ranked using infinite resources. Spammers can manipulate the algorithms by generating malicious structures in the host graph. The results of these algorithms are later compared with our proposed method.

C. IP graph

We ranked IPs using the IP graph. The IP graph is derived by reducing the host graph of IRLbot. We thought we could find higher quality hosts on higher ranked IPs. We ranked IPs using level-1 supporters(IN) and level-2 supporters(SUPP)[3] on the IP graph. The top 15 IP addresses based on IN and SUPP is shown in Table VIII. We found that there were few spam IPs ranked higher. On further analysis we found that one of them was a parking services owned by GoDaddy, a hosting providers. Hosting IPs had high IN and SUPP on the IP graph due to the IP diversity of the sites hosted on them. Hence, we found that the IP graph was not very useful to identify quality hosts.

D. Heterogenous graphs

Spammers could generate an infinite number of hostnames to support their hosts. It would be harder for them to get a large number of finite resources to support them. For example, it would be harder to get a large amount of PLDs supporters at level-2 due to the cost involved in buying PLDs. We reduce the $host \leftarrow host$ ingraph to $host \leftarrow PLD$ graph. We merge this graph with the PLD ingraph as shown in Fig. 1. We then calculated SUPP on this graph. We create k partitions of the PLD ingraph and keep few of them in memory. We then read through the $host \leftarrow PLD$ graph from disk to find level-2 supporters in the partitions in memory and merge the counts across different partitions to find the total PSUPP value.

$$PSUPP(j) = \sum_{p=1}^{k} \sum_{i=1}^{n_p} 1_{d(i,j)=2}, \tag{5.2}$$

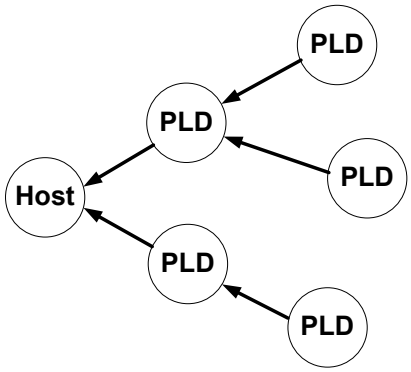


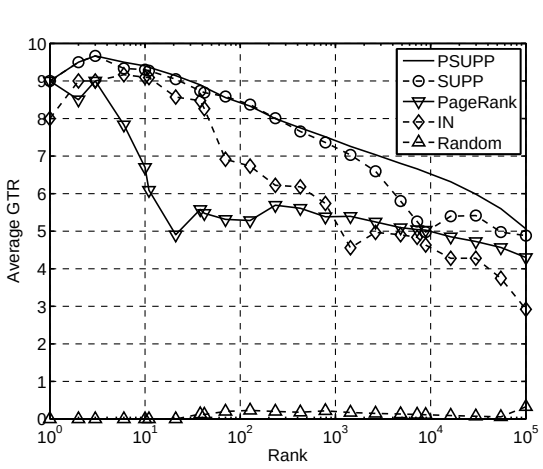
Fig. 1. Host-PLD-PLD ingraph.

where n_p is the number of nodes in partition p. The number of disk reads required is $O(k(G+P))$ where G is the size of the $host \leftarrow PLD$ graph and P is the maximum size of a partition.

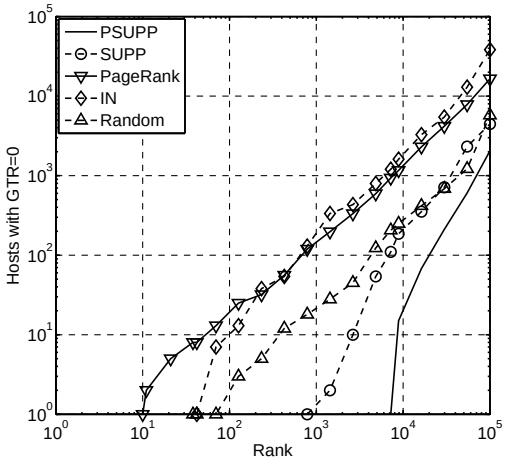
E. Analysis

We first use Google Toolbar Ranks(GTR) to compare rankings similar to the study in [3]. We compare all the algorithms with a randomly generated ranking from the graph. We find the GTRs for the top 100,000 hosts in PageRank, SUPP, IN, Random and PSUPP. The top 10 sites from the algorithms are tabulated in Table IX. PageRank had a lot of spam pages like *information.com* at the top of the ranking. IN was better for the top 10 hosts but had a lot of spam as seen next during manual analysis. SUPP and PSUPP both also perform well for the top 10 hosts. We plot the average GTR for all hosts that have a GTR value in Fig. 2(a). The GTR values of hosts without a GTR or with zero GTR values are not counted towards the average. We see that PSUPP has the highest overall average GTR. SUPP on the hostgraph performs

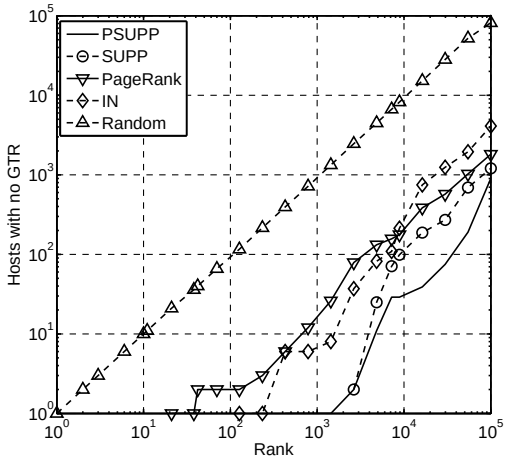
consistently and close to PSUPP, but drops after 10,000 hosts. PageRank and IN both have an average GTR of around 5 after the first 1000 hosts. Their performance is inconsistent. The number of hosts which have a GTR=0 is plotted in Fig. 2(b). PSUPP performs the best here with the first host with zero GTR appears around the rank of 10,000. The number of hosts for which Google did not have a GTR is plotted in Fig. 2(c). These included hosts for which Google did not assign a GTR and also non-existent hosts. We also compare the ranking algorithms for spam by sampling hosts and manually analyzing them. We sample from each ranking the hosts that have a GTR < 4 with varying probabilities as explained next. We sample with a probability 0.5 for the hosts in the first 1000 ranks, 0.33 for hosts ranked from 1000 to 10,000 and 0.01 for the hosts from 10,000 to 100,000. This is plotted in Fig. 3. Since data from IRLbot is from the crawl in 2007, it is possible that some legitimate sites that were highly ranked in 2007, are non-existent and parked now. Thus, we separately count parking sites and the rest of spam pages. We plot the count of parked sites in Fig. 3(a) and the rest of the spam pages in Fig. 3(b). We notice that majority of the spam pages are parked sites. PSUPP also doesn't have spam till close to rank 10,000. The total projected spam in the different methods is tabulated in Table X. IN has the highest amount of spam in the top 1000, 10,000 and 100,000 hosts. This is followed by PageRank which has lot of spam between 1000 and 10,000 ranks, improves after 10,000 hosts. This can also be noticed in the average GTR plot where the GTR slightly improves for PageRank, while it drops for IN. PSUPP performs the best based in terms of both average GTR and spam found with only 3% spam in the top 100,000 hosts. SUPP has a high GTR and low spam till the top 10,000 hosts. Its GTR drops after 10,000 hosts and spam count increases.



(a) average GTR

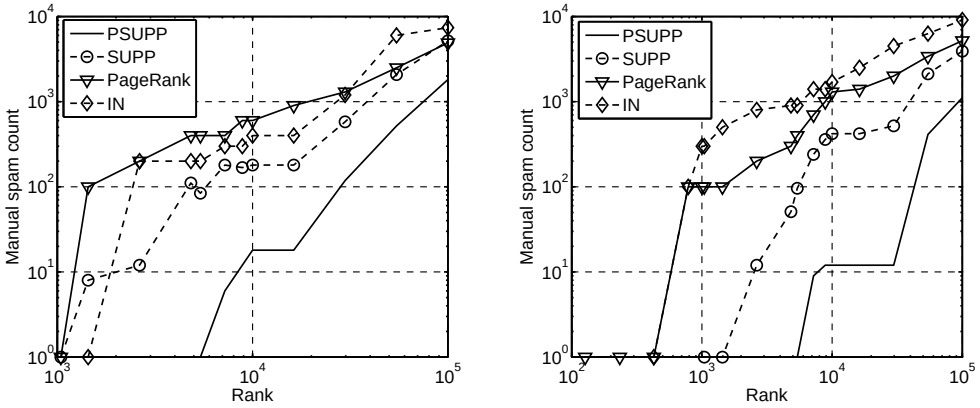


(b) GTR=0



(c) no GTR

Fig. 2. Comparison of PSUPP with existing methods.



(a) parked (b) other spam

Fig. 3. Manual spam count.

Table VIII. Top 15 IPs using IN and SUPP in the IP graph.

Rank	**IN value**		**SUPP value**	
1	208.73.210.28 (Oversee NOC)	544,535	208.73.210.28 (Oversee NOC)	2,443,090
2	192.150.18.200 (Adobe)	459,941	64.202.189.170 (GoDaddy)	2,395,103
3	64.4.11.160 (Microsoft)	438,769	216.8.179.24 (Next Dimension Inc)	2,388,060
4	74.125.227.17 (Google)	433,460	192.150.18.200 (Adobe)	2,378,203
5	64.202.189.170 (GoDaddy)	432,493	74.125.227.17 (Google)	2,374,703
6	216.8.179.24 (Next Dimension Inc)	421,693	216.8.179.24 (Next Dimension Inc)	2,388,060
7	165.91.254.15 (Akamai)	327,514	165.91.254.15 (Akamai)	2,353,005
8	208.80.152.2 (Wikipedia)	320,186	165.91.254.16 (Akamai)	2,330,865
9	165.91.254.16 (Akamai)	317,473	68.178.232.100 (GoDaddy)	2,329,213
10	68.178.232.100 (GoDaddy)	307,376	64.95.64.197 (NameMedia)	2,300,980
11	74.125.47.191 (Google)	301,484	208.80.152.2 (Wikipedia)	2,295,074
12	98.137.46.72 (Yahoo)	290,989	192.150.8.118 (Adobe)	2,283,943
13	69.43.160.145 (Trellian Pty)	275,505	74.125.47.191 (Google)	2,280,209
14	192.150.8.118 (Adobe)	264,630	98.137.46.72 (Yahoo)	2,247,380
15	64.95.64.197 (NameMedia)	262,093	69.43.160.145 (Trellian Pty)	2,239,908

Table IX. Top 10 sites ranked by different algorithms.

	PageRank		**IN**		**SUPP**		**PSUPP**	
Rank	Site	GTR	Site	GTR	Site	GTR	Site	GTR
1	go.microsoft.com	9	www.blogger.com	8	www.microsoft.com	9	www.microsoft.com	9
2	www.blogger.com	8	www.google.com	10	www.google.com	10	www.google.com	10
3	www.adobe.com	10	validator.w3.org	9	www.adobe.com	10	www.adobe.com	10
4	searchportal.information.com	5	go.microsoft.com	9	en.wikipedia.org	9	www.macromedia.com	10
5	sptc.information.com	5	www.microsoft.com	9	groups.google.com	9	www.geocities.com	9
6	www.google.com	10	www.adobe.com	10	www.geocities.com	9	www.apple.com	9
7	www.microsoft.com	9	wordpress.org	9	www.macromedia.com	10	en.wikipedia.org	9
8	www.macromedia.com	10	www.yahoo.com	9	www.amazon.com	9	www.youtube.com	9
9	www.webdiggers.net	1	www.geocities.com	9	www.myspace.com	9	www.amazon.com	9
10	www.abcsearcher.com	0	en.wikipedia.org	9	www.youtube.com	9	www.w3.org	10

Table X. Projected % spam in the top hosts.

Algorithm	top 1000	top 10,000	top 100,000
IN	30	21	16.5
PageRank	10	19	10.1
SUPP	0	0.6	9.1
PSUPP	0	0.03	2.9

CHAPTER VI

TRUE SUPPORT(TSUPP)

When we ranked hosts using PSUPP, we found several questionable sites like *www.sedo--parking.com* in the top 10K hosts. It is possible for a spammer to still manipulate PSUPP. We found that just a few links from a reputed site could boost the rankings of an obscure site significantly. It is also possible for spammers to create structures in the web to make their sites rank higher. So, we removed some edges on the host graph based on certain conditions and ranked the hosts again.

A. Removing stolen support

It is possible for a site to get a large number of supporters if it could generate links from popular sites, which can be done by adding links from blogs, forums. It is also possible that a higher ranked site inadvertently places a link to a questionable site. For example, we found in our dataset that; a link from *research.microsoft.com* to *searchportal.information.com* led to a large number of supporters for *searchportal.information.com* which is a parking service. We found that it came from a page of a research study which concluded that *searchportal.informaiton.com* was a spam page and added a link to it. It is important for highly ranked sites to set *nofollow* to the *rel* attribute of links they don't wish to endorse. We decide to count only the support from sites that truly mean to boost your ranking and which are ranked lower than it. Support to a site should also comes homogenously through its neighbors, a single link should not contribute to a significant share of the total support for a site. We compute support in this egalitarian environment and consider that this is the true support a site deserves. Our goal is identify high quality sites which would be allocated higher budgets. We observed that such sites generally have a high value

of support even after removing the links. We call this improved method as True supporters(TSUPP).

To Compute TSUPP, we remove all inlinks in the host ingraph that come from higher ranked hosts to a lower ranked host. After we remove the links, we re-rank the hosts and repeat the process again using the new ranks. We stop after the second iteration since there is not much of a difference in the graph. We then remove outlier inlinks from hosts which contribute more than ε % of the total PSUPP value to a host. We used a ε value of 5 %.

B. Removing support from link farms

It is possible that a site be promoted by spammers. A target site could achieve a high rank if it could get support from a large number of low ranked sites. A reputed site like *google.com* could also have a large number of low ranked sites. The difference for a spammer, however, is that its supporters will usually reside on a fewer IPs. Spammers own fewer IPs as they involve a significant costs and as evidenced by the high host density we observed on spam servers. We make use of the host-IP-PLD graphs to identify hosts supporting a site and residing on the same IP. So, if there are more than one host inlinks from a given IP to a host, only one randomly chosen host is retained, inlinks from all other hosts from that IP are dropped.

C. DNS blacklists

Our goal is to efficiently find the top quality sites for budget allocation. This is a small fraction of the whole collection of sites. Top sites also get support from other high quality sites, but don't depend on spam and low quality content for support. We also found that many spam servers were cohosted with such low quality content. We

propose two methods to remove the low quality sites and spam from our data set.

In the first method, we develop a new algorithm to cluster spam using DNS cohosting information and PSUPP. We first compute the average PSUPP value for each IP over the hostnames that are hosted on that IP.

$$PSUPP(i) = \frac{1}{n}\sum_{i=1}^{n} PSUPP(h_i), \tag{6.1}$$

where h_1, h_2,...,h_n are hosted on i. We use the average PSUPP(i) while adding new spam IPs. A webserver that hosts high quality websites will have a higher PSUPP(i). We create of seed set of IPs, from which we find more IPs using a graph we have constructed out of the DNS structure. The seed set is defined as below :

$$IP_H = \{i,\ \forall\ i\ \in IP : |N_H(i)| > a\ and\ PSUPP(i) < b, \} \tag{6.2}$$

where a is the threshold host density and b is the threshold average PSUPP value for an IP address. We set a to 10,000 and b to 10,000 and sampled the hosts on IP_H. We found that 99% of the hosts were spam. When sampled, we noticed that all of them were low quality sites which were unlikely to be ranked in the top 100,000 hosts. We have created a graph using DNS information. The vertices of the graph are IP addresses and authoritative IPs. The edges are between IPs and authoritative IPs. This graph can help us in finding alliances and co-hosted content. So, we perform a BFS from IP_H and add IPs which satisfy $PSUPP(i) < b$. We finally remove from the graph the hosts that are hosted on these IPs.

In our second method to remove low quality content, we define the average rank of an IP as the average of PSUPP ranks of all the hosts on that IP. We use the DNS graph as above but choose IPs who have an average PSUPP rank greater than c. After removing the edges and hosts from the host graph, we generate the new $host \leftarrow PLD$ graph and PLD ingraph and calculate PSUPP.

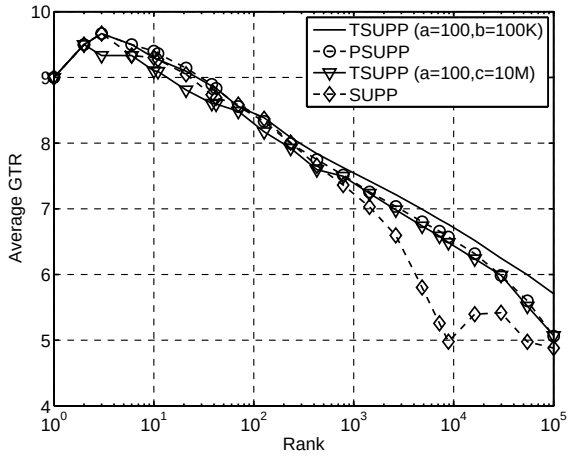
Table XI. Projected % spam in the top 100,000 hosts.

Algorithm	% spam
PSUPP	2.9
TSUPP (a=1000,b=5000)	1
TSUPP (a=100,b=50,000)	0.3
TSUPP (a=100,b=100,000)	0.3

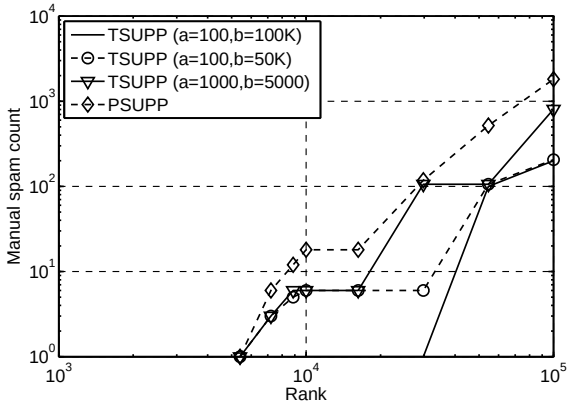
D. Analysis

The average GTR is plotted in Fig. 4(a). We observed that TSUPP significantly improved the average GTR of PSUPP when using blacklists from the DNS graph. The total number of parked sites and other spam sites are plotted in Fig. 4(b) and Fig. 4(c) respectively. It is seen that the amount of spam in the top 100,000 hosts of TSUPP could be controlled by the parameters chosen for the blacklisted hosts. The total projected spam in the different methods is tabulated in Table XI.

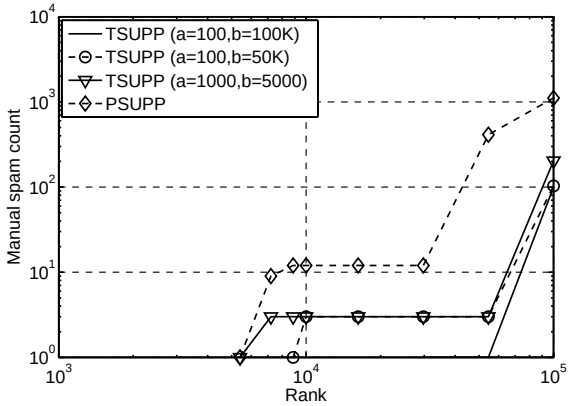
We find that there is around 0.3% spam in the top 100,000 hosts of TSUPP for a=100 and b=100,000. We still found a few parked sites in the top 100,000. This is because the IP addresses of these sites were not in our dataset. Their IP addresses had changed after our DNS resolutions. Parked sites though will not be a problem when performing a fresh crawl and elimination of other types of spam is more crucial. We also compare PSUPP and TSUPP for a=100 and b=100,000 to see how TSUPP modifies PSUPP. This is plotted in Fig. 5. We observe that there is not much change in the top 1000 hosts, but few of the spam hosts are moved down the ranking in TSUPP.



(a) average GTR



(b) parked



(c) other spam

Fig. 4. Comparison of TSUPP with other algorithms.

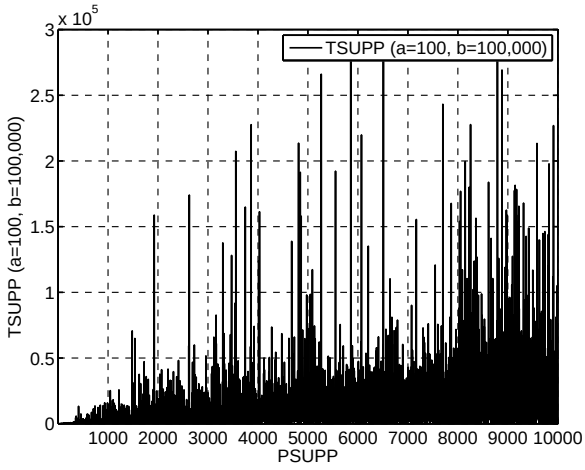


Fig. 5. Comparison of TSUPP with PSUPP.

CHAPTER VII

CONCLUSION AND FUTURE WORK

It is a challenging problem to rank an infinite resource like hostname, since spammers can manipulate ranking algorithms. In this paper, we have proposed two algorithms to rank hosts. TSUPP performs efficiently compared to existing ranking algorithms in identifying the top hosts and eliminating spam. We first tried existing algorithms PageRank, IN and SUPP and found that they have a significant amount of spam in the top 100,000 hosts. We propose a novel method to generate heterogenous graphs with finite resources like PLDs to rank hosts. The basic version of this called PSUPP performs better than all existing methods in terms of average GTR and the manual spam count. There is around 3% spam in the top 100,000 hosts. We then propose another ranking algorithms called TSUPP which applies a variety of rules to remove hosts and links from the host graph. The links are based on true egalitarian support and host-IP-PLD graph is used to merge multiple hosts on the same IP. We also apply DNS blacklists to remove certain low quality IPs from the graph. The improved version of PSUPP performs significantly better and remove most of the spam in PSUPP, with only 0.3% spam in the top 100,000 hosts.

We have also proposed a method to efficiently parallelize computation of SUPP on large graphs by creating partitions. We implemented it and could be used to compute SUPP values in real-time. By considering only the top 100,000 hosts which have negligible spam in TSUPP, we could save a lot of overhead in future crawls.

The algorithms proposed for the host graph can be extended to the PLD graph and be compared to existing algorithms for ranking PLDs. We found that SUPP performed poorly for hosts and IPs. It ranked spam IPs and hosts in the top of the ranking. However, SUPP on the PLD graph was found to have worked well. We can

compare the position of hosts in its ranking of PLD, IP and hosts to find anomalies. If there is an inconsistency, a host could be marked as suspicious.

REFERENCES

[1] H. Lee, X. W. D. Leonard, and D. Loguinov, "IRLbot: Scaling to 6 Billion Pages and Beyond," in *Proc. World Wide Web Conference*, pp. 427–436, April 2008.

[2] L. Page, S. Brin, R. Motwani, and T. Winograd, "The Pagerank Citation Ranking: Bringing Order to the Web.," *Technical Report* 1999-66, *Stanford InfoLab*, 1999.

[3] C. Sparkman, H.-T. Lee, and D. Loguinov, "Agnostic Topology-Based Spam Avoidance in Large-Scale Web Crawls," in *Proc. IEEE INFOCOM*, pp. 811–819, Oct. 2011.

[4] D. Fetterly, M. Manasse, and M. Najork, "Spam, Damn Spam, and Statistics: Using Statistical Analysis to Locate Spam Web Pages," in *Proc. ACM Workshop on the Web and Databases*, pp. 1–6, June 2004.

[5] Z. Gyngyi and H. Garcia-molina, "Web Spam Taxonomy," in *Proc. Workshop on Adversarial Information Retrieval on the Web*, pp. 39–47, May 2005.

[6] J. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," *Journal of ACM*, vol. 46, pp. 604–632, Sept. 1999.

[7] Z. Gyöngyi and H. Garcia-Molina, "Link Spam Alliances," in *Proc. International Conference on Very Large Data Bases*, Aug. 2005.

[8] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen, "Combating Web Spam with Trustrank," in *Proc. International Conference on Very Large Data Bases*, pp. 576–587, Aug. 2004.

[9] A. Benczr, K. Csalogny, and T. Sarls, "SpamRank – Fully Automatic Link Spam Detection," in *Proc. Workshop on Adversarial Information Retrieval on the Web*, pp. 25–38, May 2005.

[10] B. Wu, V. Goel, and B. Davison, "Topical Trustrank: Using Topicality to Combat Web Spam," in *Proc. World Wide Web Conference*, pp. 63–72, May 2006.

[11] J. Caverlee and L. Liu, "Countering Web Spam with Credibility-Based Link Analysis," in *Proc. ACM Symposium on Principles of Distributed Computing*, pp. 157–166, Aug. 2007.

[12] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates, "Using Rank Propagation and Probabilistic Counting for Link-Based Spam Detection," in *Proc. Knowledge Discovery on the Web*, Aug. 2006.

[13] L. Nie, B. Wu, and B. Davison, "Winnowing Wheat from the Chaff: Propagating Trust to Sift Spam from the Web," in *Proc. ACM SIGIR*, pp. 869–870, July 2007.

[14] B. Wu and B. D. Davison, "Identifying Link Farm Spam Pages," in *Proc. World Wide Web Conference*, pp. 820–829, May 2005.

[15] B. Wu and K. Chellapilla, "Extracting Link Spam using Biased Random Walks from Spam Seed Sets," in *Proc. Workshop on Adversarial Information Retrieval on the Web*, pp. 37–44, May 2007.

[16] Z. Gyongyi and H. Garcia-molina, "Link Spam Detection Based on Mass Estimation," in *Proc. International Conference on Very Large Data Bases*, pp. 439–450, Sept. 2006.

[17] H. Saito, M. Toyoda, M. Kitsuregawa, and K. Aihara, "A Large-Scale Study of Link Spam Detection by Graph Algorithms," in *Proc. Workshop on Adversarial*

Information Retrieval on the Web, pp. 45–48, May 2007.

[18] Y. Chung, M. Toyoda, and M. Kitsuregawa, "A Study of Link Farm Distribution and Evolution Using a Time Series of Web Snapshots," in *Proc. Workshop on Adversarial Information Retrieval on the Web*, pp. 9–16, April 2009.

[19] B. Wu and B. Davison, "Undue Influence: Eliminating the Impact of Link Plagiarism on Web Search Rankings," in *Proc. ACM Symposium on Applied Computing*, pp. 1099–1104, April 2006.

[20] Y. Dourisboure, F. Geraci, and M. Pellegrini, "Extraction and Classification of Dense Implicit Communities in the Web Graph," *Proc. ACM Trans. Web*, vol. 3, pp. 7:1–7:36, April 2009.

[21] X. Qi, L. Nie, and B. Davison, "Measuring Similarity to Detect Qualified Links," in *Proc. Workshop on Adversarial Information Retrieval on the Web*, pp. 49–56, May 2007.

[22] A. Benczur, K. Csalogany, and T. Sarlos, "Link-Based Similarity Search to Fight Web Spam," in *Proc. Workshop on Adversarial Information Retrieval on the Web*, pp. 9–16, Aug. 2006.

[23] D. Zhou, C. Burges, and T. Tao, "Transductive Link Spam Detection," in *Proc. Workshop on Adversarial Information Retrieval on the Web*, pp. 21–28, May 2007.

[24] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates, "Link-Based Characterization and Detection of Web Spam," in *Proc. Workshop on Adversarial Information Retrieval on the Web*, pp. 1–8, Aug. 2006.

[25] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri, "Know your Neighbors: Web Spam Detection using The Web Topology," in *Proc. ACM SIGIR*, pp. 423–430, July 2007.

[26] S. Webb, J. Caverlee, and C. Pu, "Predicting Web Spam with HTTP Session Information," in *Proc. ACM Conference on Information and Knowledge Management*, pp. 339–348, Oct. 2008.

[27] S. Yadav, A. Reddy, A. Reddy, and S. Ranjan, "Detecting Algorithmically Generated Malicious Domain Names," in *Proc. ACM Internet Measurement Conference*, pp. 48–61, Nov. 2010.

[28] J. Hirai, S. Raghavan, H. Garcia-Molina, and H. Paepcke, "WebBase: A Repository of Web Pages," in *Proc. World Wide Web Conference*, pp. 277–293, Feb. 2000.

[29] T. Jinmei and P. Vixie, "Practical Approaches for High Performance DNS Server Implementation with Multiple Threads," in *Proc. Workshop on Internet Technology* , Nov. 2005.

[30] T. Haveliwala, "Efficient computation of pagerank," *Technical Report* 1999-31, *Stanford InfoLab*, 1999.

VITA

Siddhartha Sankaran Mathiharan received his Bachelor of Technology (B.Tech.) in Computer Science and Engineering from the National Institute of Technology Trichy in Tiruchirappalli, India, in May 2008. He completed his Master of Science (M.S.) in Computer Science at Texas A&M University and graduated in December 2011.

His research interests include designing and building large-scale systems for measurements on the Internet, data mining and distributed systems . He may be contacted at:

Siddhartha Sankaran Mathiharan
c/o Department of Computer Science and Engineering
Texas A&M University
College Station
Texas - 77843-3128

The typist for this thesis was Siddhartha Sankaran Mathiharan.